

**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims:**

1. (currently amended) A method for troubleshooting a computer system, comprising:

receiving, at a location remote to the computer system, a request for problem analysis for the computer system in response to an occurrence of a problem;

receiving data corresponding to the computer system comprising a core file from the computer system and supplemental configuration information;

accessing a knowledge store storing phases, wherein each phase includes a set of scripts adapted to identify previously identified computer problems;

performing a phased analysis of the computer system data comprising completing at least two of the phases from the knowledge store; and

generating an output report including a list of the computer problems identified during the performing[.];

wherein each of the phases includes an optimization mechanism operable such that the phases executed in the performing is less than the set of the phases, wherein the optimization mechanism functions to check a list of static preconditions and to compare the static preconditions to the received computer system data; and

wherein the static preconditions include variables selected from the group consisting of operating system identifications, operating system releases, software packages, software packages releases, system architectures, platforms, patch levels, and panic strings.

2. (original) The method of claim 1, wherein the core file comprises a memory image created by the computer system in response to a kernel error, a hardware error, an application error, or other computer system error.

Claims 3 and 4 (canceled)

5. (original) The method of claim 1, wherein the data receiving includes communicating with the computer system from the remote location over a communications network.

6. (original) The method of claim 1, wherein the phased analysis performing includes parsing the computer system data into a source independent format and wherein the parsed computer system data is used as input for subsequent ones of the analysis phases.

7. (original) The method of claim 1, wherein the phased analysis performing comprises a sequential analysis using the analysis phases to process a memory image from the computer system to create at least one corrective action for inclusion in the output report.

8. (original) The method of claim 7, wherein the analysis phases are selected from the group consisting of a hardware error scan, a bug analysis, a user core file analysis, a kernel core file analysis, a parse of the computer system data into context free format, a check for bad patches, a check for software configuration and version, a check for hardware configuration and version, a software error scan, an infodoc check, a faq check, an srdb check, an stb/white paper check, retrieval of down revision patches, a security issue check, a system health check, a storage-related check, a platform analysis, a performance analysis, a kernel configuration check, and third party checks.

9. (original) The method of claim 7, wherein the analysis phases each generate an intermediate output report for use by later performed ones of the analysis phases and for inclusion in the output report.

Claims 10-12 (canceled)

13. (currently amended) A method for reactively troubleshooting and proactively controlling problems on a computer system, comprising:

- collecting data corresponding to the computer system;
- accessing a knowledge store of previously identified computer system problems;
- parsing the collected data into a context-free, language-independent format;
- building a list of analysis phases to run on the collected data;
- running each phase in the list using the collected data; and
- generating an analysis report based on output data from at least some of the analysis phases run[.];

wherein each phase comprises a plurality of scripts and wherein the running of each phase includes first performing a static precondition check for the phase to determine if a subset of the scripts applies to the computer system to improve run times by reducing the scripts executed during running.

14. (canceled)

15. (currently amended) The method of claim [[14]] 13, wherein the scripts comprise testing the collected data for an instance of a problem, issuing a reactive output when the instance is found in the testing, and issuing a proactive output when the instance is not found and the problem is determined to possibly occur within the computer system.

16. (original) The method of claim 15, wherein the reactive output and the proactive output includes supplemental data for each of the problems comprising type, severity, comment, and derived information.

17. (original) The method of claim 16, wherein a type is proactive, reactive, or not applicable.

18. (original) The method of claim 16, wherein a severity ranges from most to least severe and is selected from the group consisting of data error, system hang, system panic, device hang, application hang, application crash, error warning message, system maintenance issue, administration issue, informational message, and impaired functionality.

19. (original) The method of claim 16, wherein a comment is sequential text selected to be useful in interpreting the problems.

20. (original) The method of claim 16, wherein a derived information includes information extracted from a service repository.

21. (new) A method for troubleshooting a computer system, comprising:

receiving, at a location remote to the computer system, a request for problem analysis for the computer system in response to an occurrence of a problem;

receiving data corresponding to the computer system comprising a core file from the computer system and supplemental configuration information;

accessing a knowledge store storing phases, wherein each phase includes a set of scripts adapted to identify previously identified computer problems;

performing a phased analysis of the computer system data comprising completing at least two of the phases from the knowledge store; and

generating an output report including a list of the computer problems identified during the performing;

wherein the phased analysis performing comprises a sequential analysis using the analysis phases to process a memory image from the computer system to create at least one corrective action for inclusion in the output report; and

wherein the analysis phases are selected from the group consisting of a hardware error scan, a bug analysis, a user core file analysis, a kernel core file analysis, a parse of the computer system data into context free format, a check

for bad patches, a check for software configuration and version, a check for hardware configuration and version, a software error scan, an infodoc check, a faq check, an srdb check, an stb/white paper check, retrieval of down revision patches, a security issue check, a system health check, a storage-related check, a platform analysis, a performance analysis, a kernel configuration check, and third party checks.

22. (new) An apparatus for troubleshooting a computer system, comprising:

means for receiving a request for problem analysis for the computer system in response to an occurrence of a problem;

means for receiving data corresponding to the computer system comprising a core file from the computer system and supplemental configuration information;

means for accessing a knowledge store storing phases, wherein each phase includes a set of scripts adapted to identify previously identified computer problems; and

means for performing a phased analysis of the computer system data comprising completing at least two of the phases from the knowledge store;

wherein each of the phases includes an optimization mechanism operable such that the phases executed in the performing is less than the set of the phases, wherein the optimization mechanism functions to check a list of static preconditions and to compare the static preconditions to the received computer system data; and

wherein the static preconditions include variables selected from the group consisting of operating system identifications, operating system releases, software packages, software packages releases, system architectures, platforms, patch levels, and panic strings.

23. (new) A method for reactively troubleshooting and proactively controlling problems on a computer system, comprising:  
receiving data corresponding to the computer system;  
receiving previously identified computer system problems;  
building a list of analysis phases to run on the collected data; and  
running each phase in the list using the collected data;  
wherein each phase comprises a plurality of scripts and wherein the running of each phase includes first performing a static precondition check for the phase to determine if a subset of the scripts applies to the computer system to improve run times by reducing the scripts executed during running.

24. (new) The method of claim 23, wherein the scripts comprise testing the collected data for an instance of a problem, issuing a reactive output when the instance is found in the testing, and issuing a proactive output when the instance is not found and the problem is determined to possibly occur within the computer system.

25. (new) An apparatus for reactively troubleshooting and proactively controlling problems on a computer system, comprising:  
means for receiving data corresponding to the computer system;  
means for receiving previously identified computer system problems;  
means for building a list of analysis phases to run on the collected data;  
and  
means for running each phase in the list using the collected data;  
wherein each phase comprises a plurality of scripts and wherein the running of each phase includes first performing a static precondition check for the phase to determine if a subset of the scripts applies to the computer system to improve run times by reducing the scripts executed during running.